

MULTILEVEL SECURITY



What Is Multilevel Security?

- System that is able to process data at different levels of sensitivity (different security levels) is called multilevel secure
- Provides access to users with different security clearances
- Prevents users from obtaining unauthorized access to information

Recap on Security Policy



- Threat model – what can happen?
- Security policy – what we want our system to do?
- Security target – how to do it?

Classification

- Label attached to a document
- Unclassified – public information
- Confidential – for internal use
- Secret – compromise could cost lives
- Top secret – compromise could cost many lives

Clearance



- Determines what levels of classification a person can access
- Different levels of clearances usually involve different kinds of checks

Security Policy

- Person can read document if his security clearance is at least as high as the document's classification
 - ▣ Information can only flow upward
 - ▣ Authorized person can declassify document, usually after sanitizing it
- Document-handling rules
 - ▣ e.g. secret documents must be kept in safe, rooms must be guarded, photocopiers prohibited etc.

Bell-LaPadula Model

- Developed in 1973
- **Simple security property** – no process may read data at a higher level
 - ▣ No read up
- ***-property** – no process may write data to a lower level
 - ▣ No write down

The *-Property

- The *-property is an important innovation designed to counter malicious software
- Otherwise:
 - ▣ Low user manages to execute program at high level
 - ▣ Program copies file to low level

Mandatory Access Control



- System enforces security policy independently of user actions
- Alternative is discretionary access control, which allows user to control permissions

Implementing MLS

- Reference monitor – supervises all accesses to files (OS calls) and enforces the security policy
- Pump – use separate databases for different levels of security. Use pump to copy information from Low to High

About Bell-LaPadua

- Simple
- Allows formal verification of systems
- Requires tranquility property
 - ▣ Security labels never change during system operation
 - ▣ Prevents situation where user asks the administrator to temporarily declassify file from High to Low

About Bell-LaPadua (2)

- Processes usually start at Low level
- When reading file with higher label, process'es level is upgraded
- This makes it impossible to e.g., continue writing previously opened temporary file

Biba Model

- Deals with integrity
- Dual to the Bell-LaPadua model
- Process with High level can write to Low
- Process with Low level cannot write to High
- Suitable when you want to prevent corruption of data by untrusted processes
 - ▣ For example, system configuration

Problems with MLS



- Composability – composition of secure components can lead to insecure systems
- Especially so when feedback loops can occur

Covert Channels

- High programs cannot write to Low
 - ▣ At least not directly
- However, they can smuggle information out using **covert channel**
- Usually based on a shared resource

Covert Channel Examples

- Temporary files
- Resource load/timing
 - Network
 - Memory
 - Processor
 - Hard disk
- Using steganography

Covert Channels (2)

- They are unavoidable
- The best bet is to limit bandwidth
 - ▣ Degrades system performance
 - ▣ Hard to do on high-bandwidth channel

MLS Issues

- Polyinstantiation
 - ▣ Present in MLS databases
 - ▣ What happens if High and Low users want to create same object?
- Blind write up
- Classification of data
 - ▣ Is amount of boot polish in a base classified?
- Often hard to use